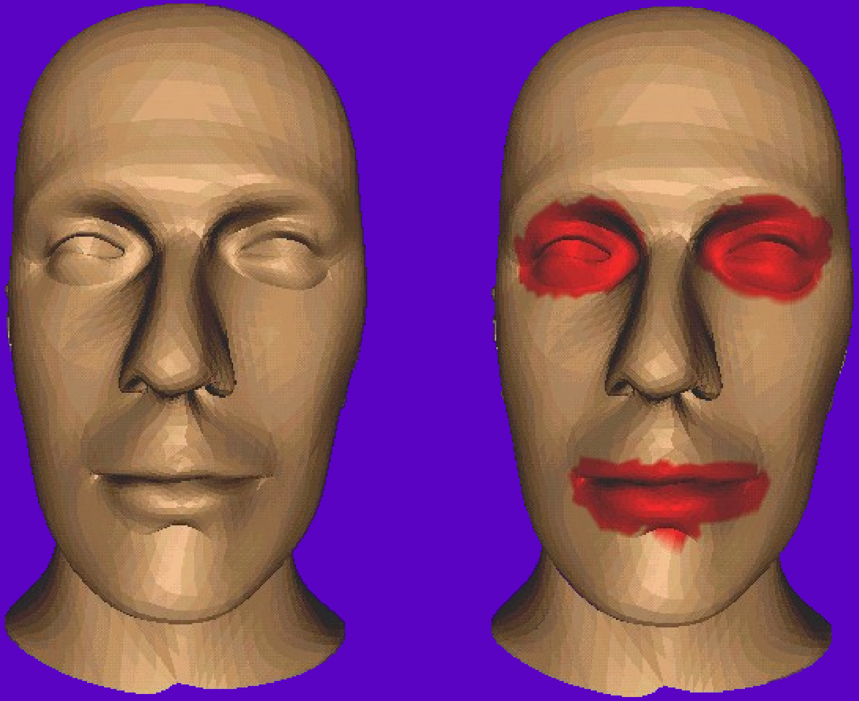
User-Guided Simplification

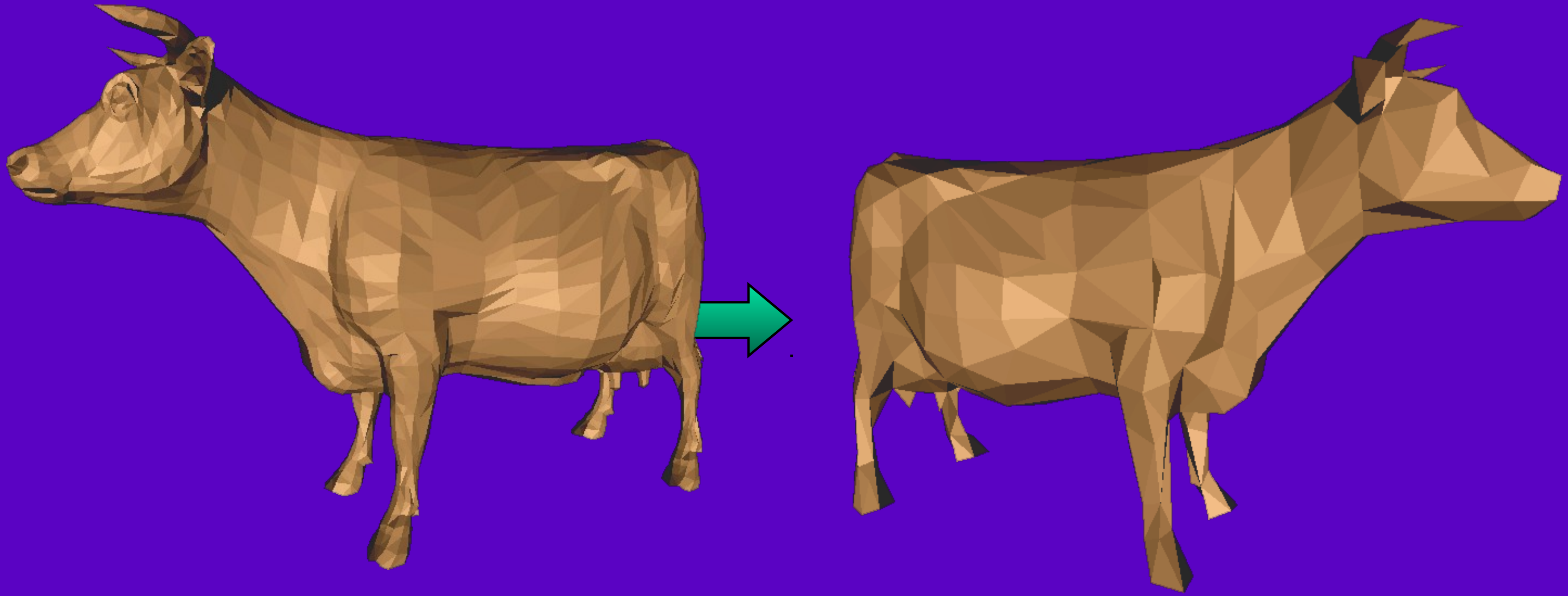


Youngihn Kho

Michael Garland

University of Illinois at Urbana–Champaign

Surface Simplification



5,804 faces

780 faces

Previous simplifications

Most of previous algorithms are fully automatic

- *generally work well, but*
- *weak at higher-level semantics*
- *user-guidance can improve the quality*

Semi-automatic methods

- Z***eta*** *(Cignoni* et al. *'97)*
- ***Semisimp*** *(Li and Watson '01)*
- User-Controlled Creation of Multiresolution Meshes

  (*Pojar and Schmalstieg '03*)

User Guidance can be Useful

We propose a user-guided simplification

- directly guide simplification processes
- users freely interact at any level

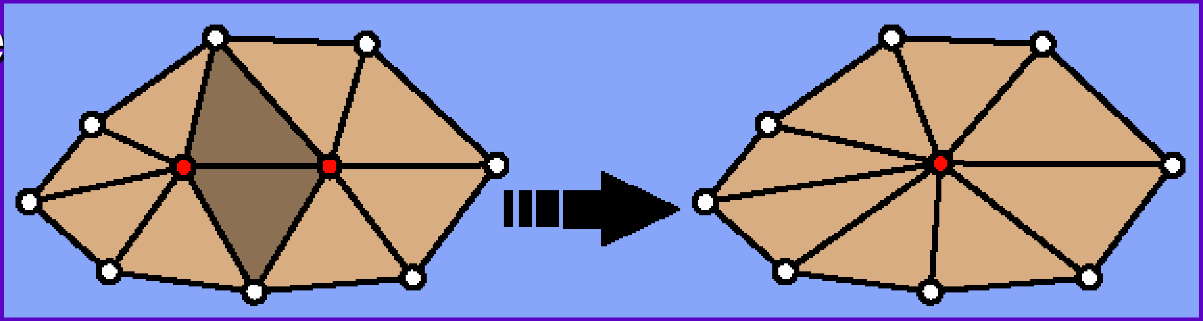
- Built *on top of* quadric-based simplification (Garland and Heckbert ’97)

Demo #1
Face model

Iterative Edge Contraction

Starting from the original model, iteratively

- rank all edges with some cost metric
- contract minimum cost edge
- remove degenerate faces
- update neighboring edges
- ge



Quadric-Error Metric

Given a plane, a quadric Q is defined as

$$Q = (\mathbf{A}, \mathbf{b}, c) = (\mathbf{n}\mathbf{n}^{\top}, d\mathbf{n}, d^{<})$$

Squared distance of a point to the plane is

$$Q(\mathbf{v}) = \mathbf{v}^{\top}\mathbf{A}\mathbf{v} + <\mathbf{b}^{\top}\mathbf{v} + c$$

Sum of quadrics represents a set of planes

$$\sum_{i} (\mathbf{n}_i^{\top}\mathbf{v} + d)^{<} = \sum_{i} Q_i(\mathbf{v}) = \left(\sum_{i} Q_i\right)(\mathbf{v})$$

Quadric-Error Metric

Each edge has an associated quadric

- sum of quadrics for its two vertices
- find a vertex **v*** minimizing *Q*(**v***)

$$\nabla Q(\mathbf{v}^*) = : \quad \Rightarrow \quad \mathbf{A}\mathbf{v}^* = -\mathbf{b}$$

After the edge contraction

- the vertex **v*** accumulates the associated quadrics

$$Q_{\mathbf{v}^*} = Q_{\mathbf{v}_i} + Q_{\mathbf{v}_j}$$

How to Guide Simplification: Manipulate Quadrics

In the quadric-based algorithm

- contraction order and optimal positions are crucial
- quadrics determine both

We manipulate quadrics in two main ways

- weighting quadrics
- adding constraint quadrics

Weighting Quadrics: Control Contraction Costs

We weight quadrics

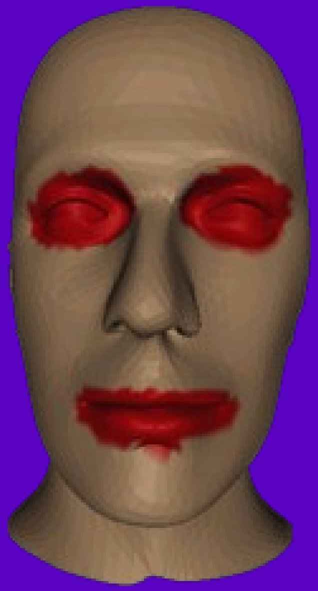
$$Q_i \leftarrow w_i Q_i$$

Heavy weights are applied

- on feature areas
- increase contraction costs

Also changes optimal positions

- a desirable property

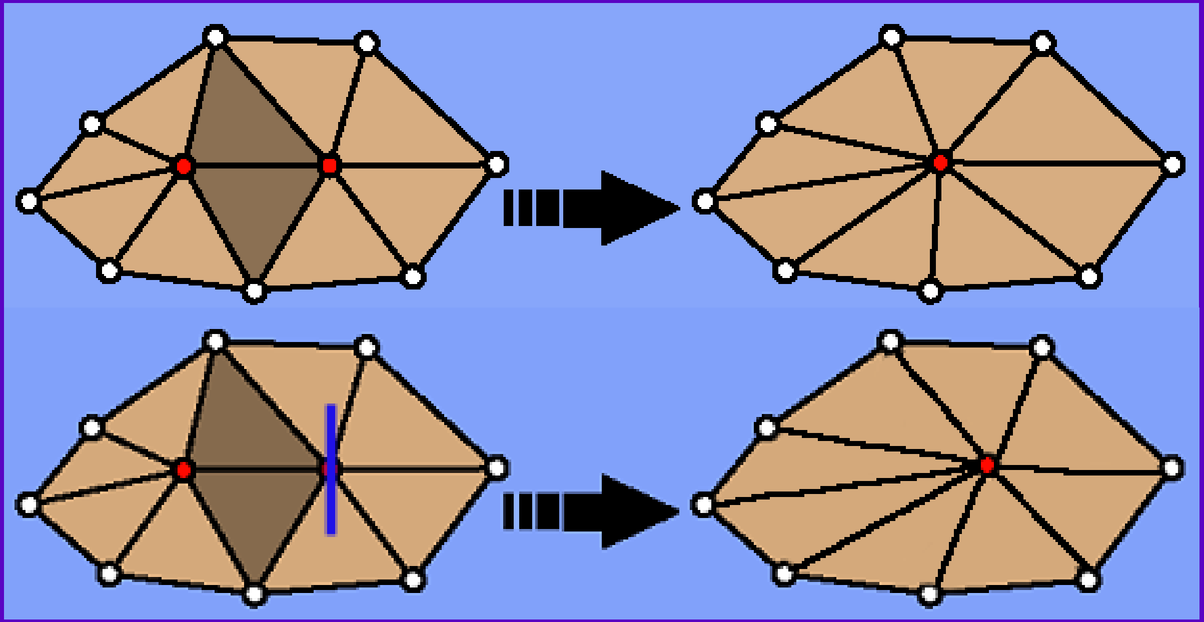


Feature areas are painted

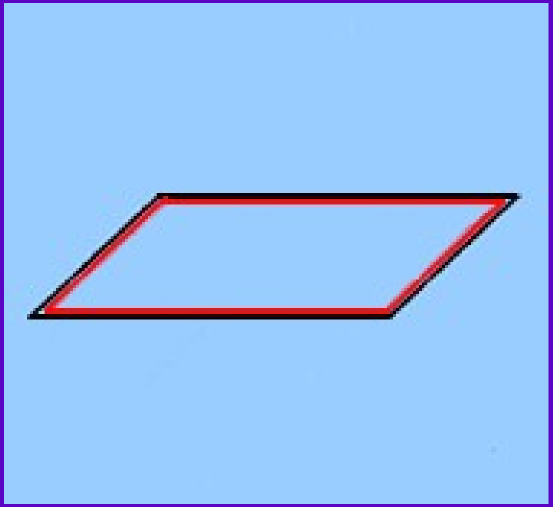
Constraint Quadrics:
Control Optimal Positions

We can define constraint planes

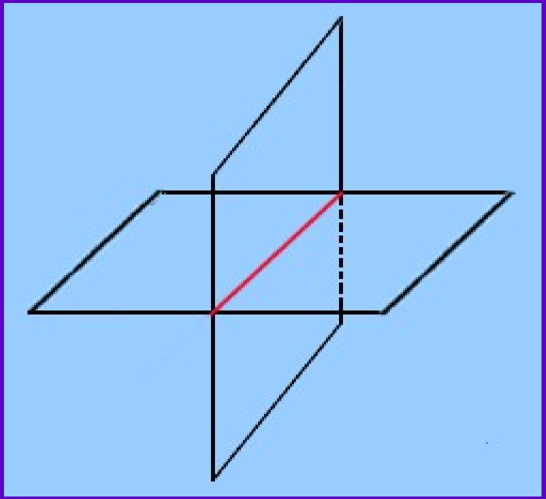
- add their quadrics to appropriate vertices
- bias optimal positions
- increase contraction costs -> store separately



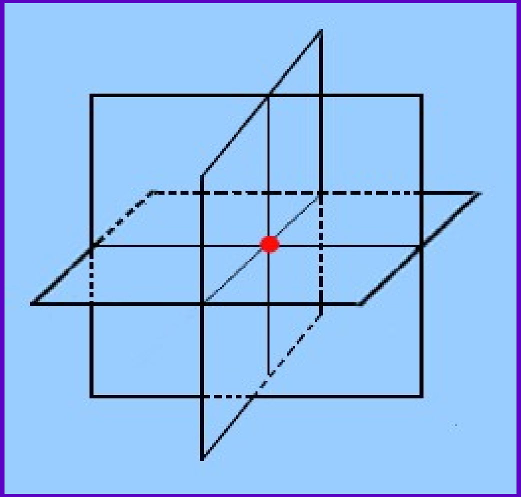
We Propose Three types of Constraint Quadrics



Plane Constraints

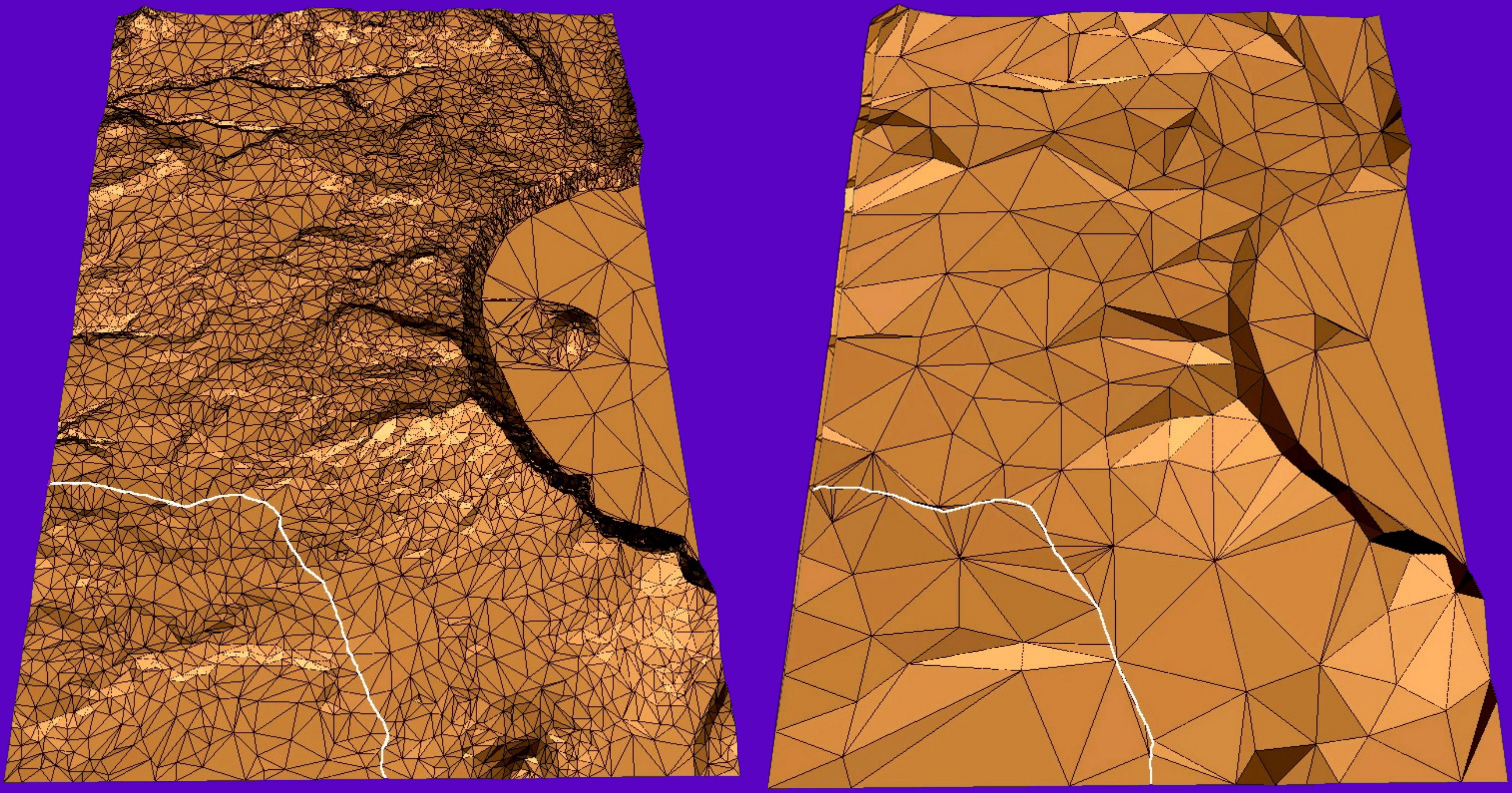


Contour Constraints

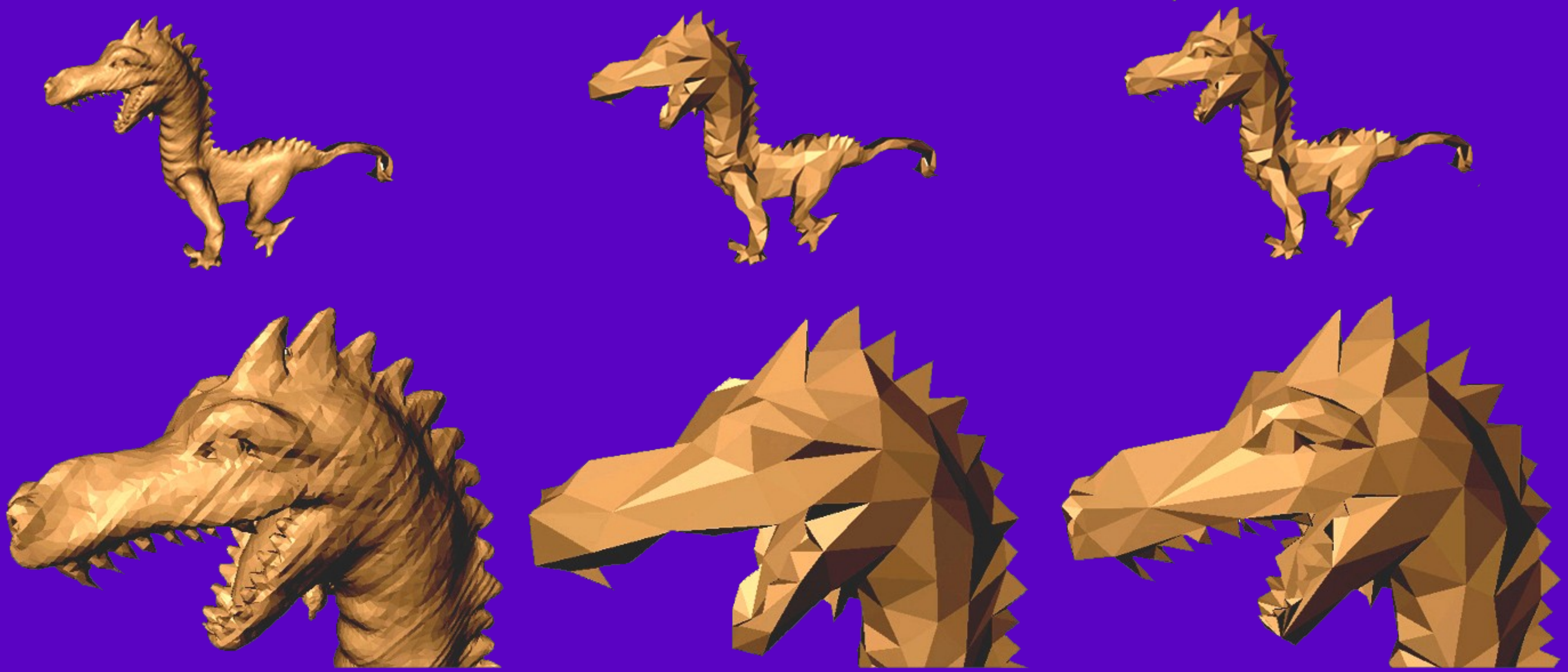


Point Constraints

Example : Contour Constraint



Example : Point Constraint



We Need:
New Propagation Rules

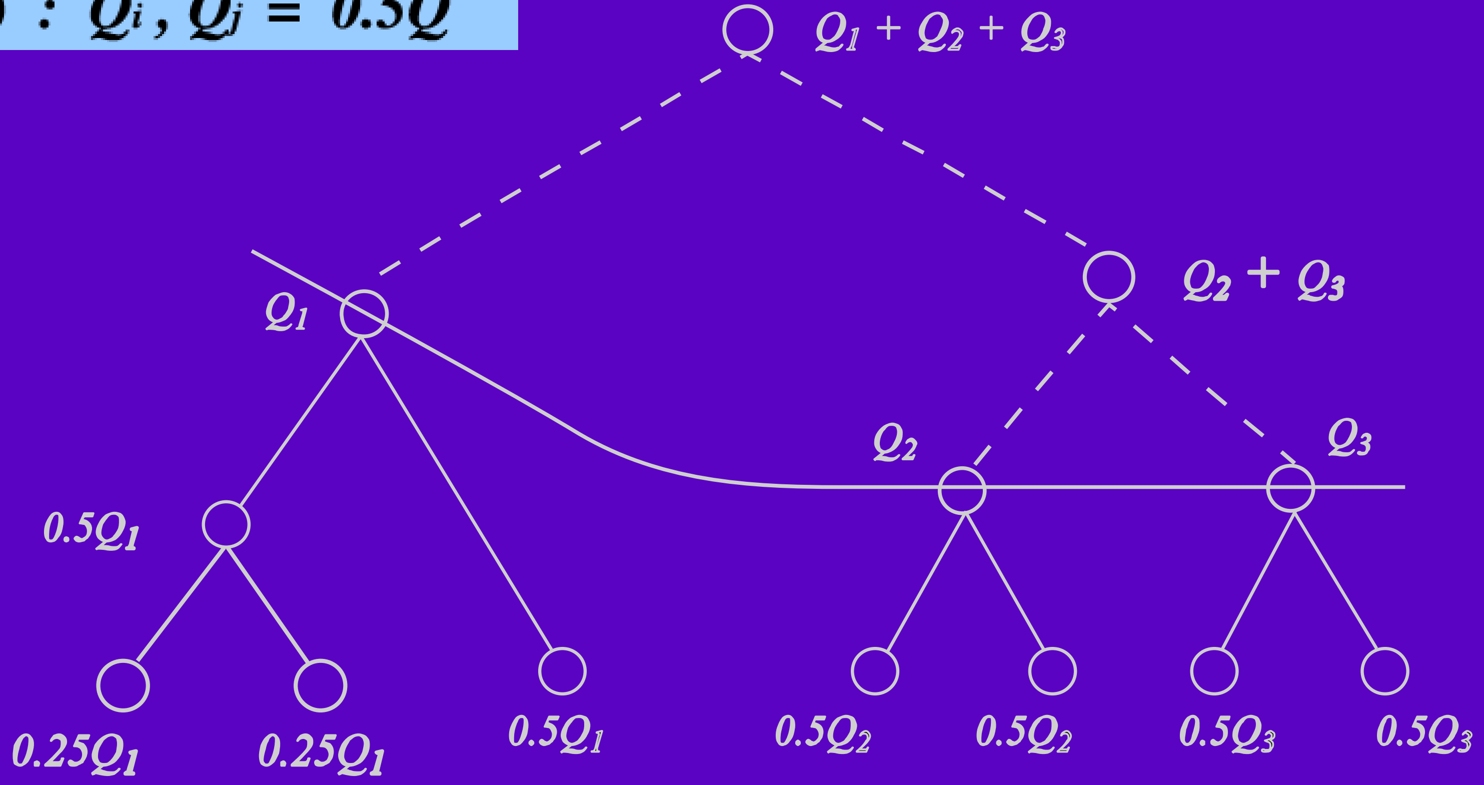
Users want to freely interact at any level

- affect both simplification and refinement process
- vertex tree structures may be changed
- constraint quadrics and *representative weights* should be appropriately propagated.

Propagation: Constraint Quadrics

$$(V_i, V_j) \rightarrow V^*: Q^* = Q_i + Q_j$$

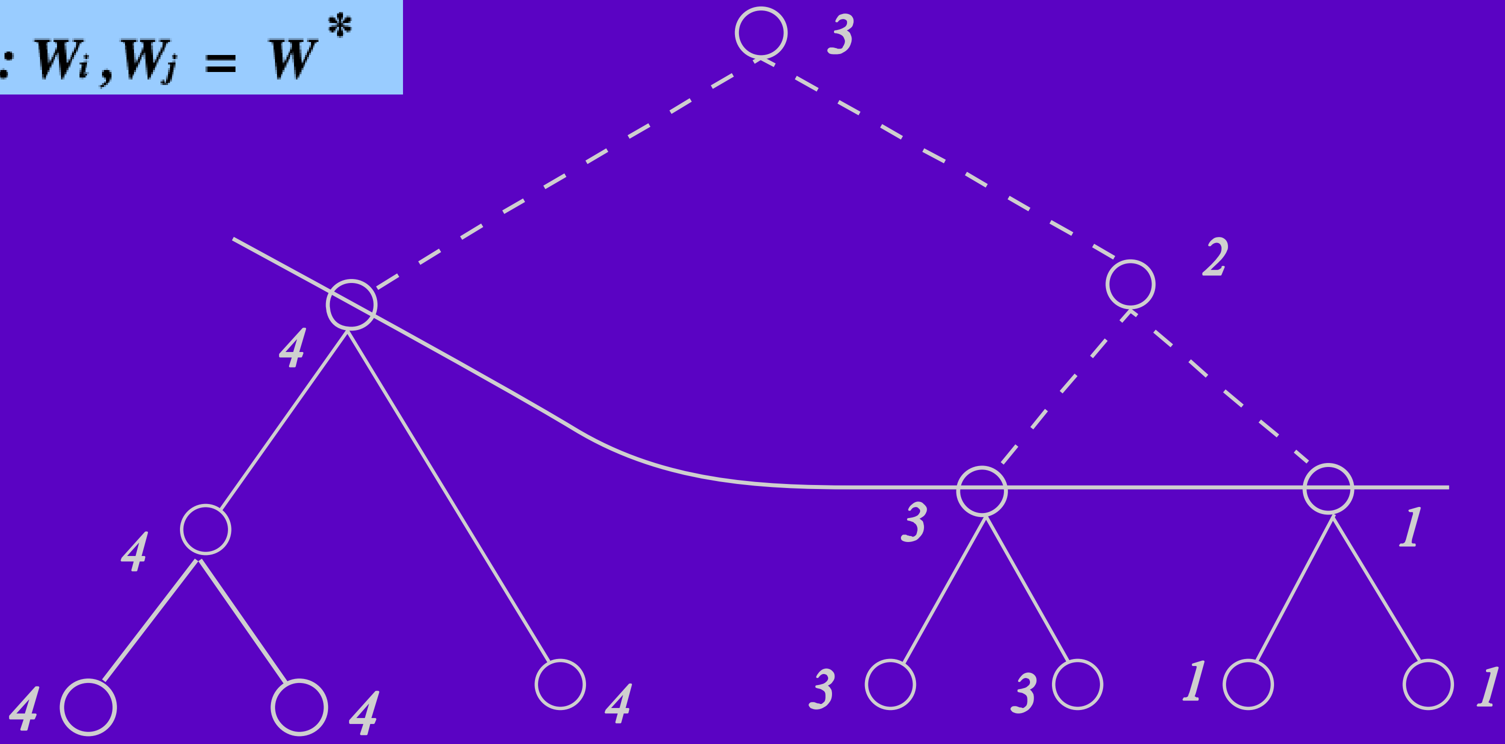
$$V^* \rightarrow (V_i, V_j) : Q_i, Q_j = 0.5Q^*$$



Propagation:
Representative Weight

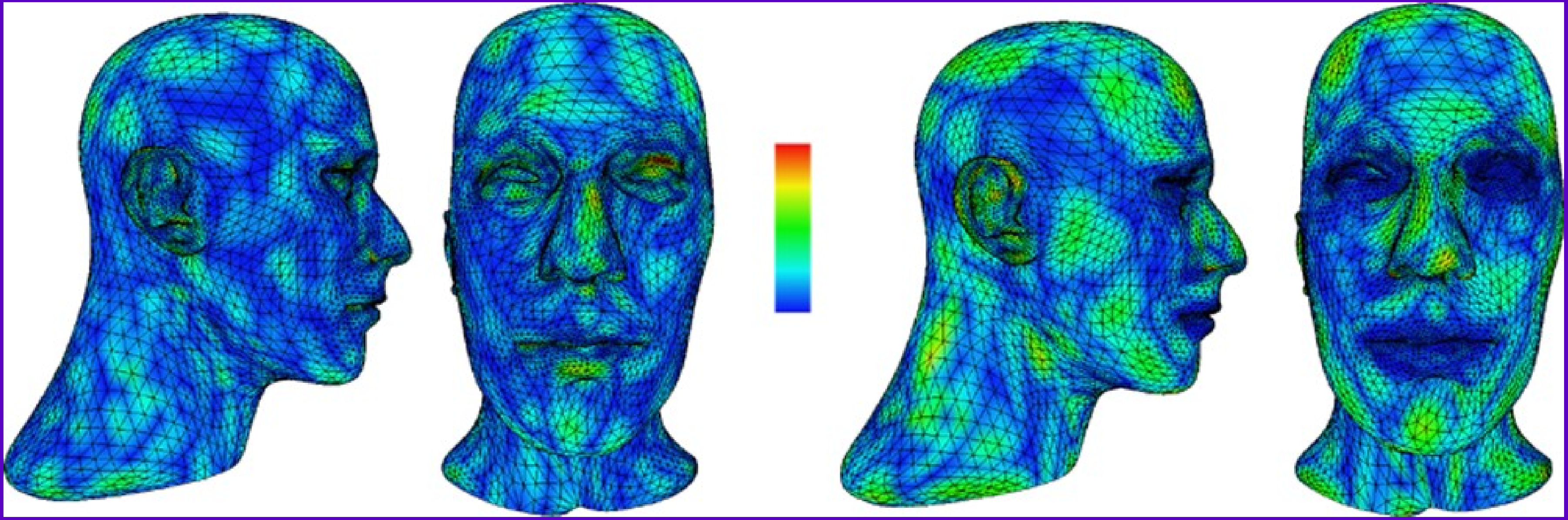
$$(V_i , V_j) \rightarrow V^* : \; W^* = (W_i + W_j) / 2$$

$$V^* \rightarrow (V_i , V_j) : W_i , W_j = W^*$$



Demo #2
Dragon model

Error Analysis:
Relative Error Distribution



Fully automatic

User-guided

Conclusion

New interactive simplification system

- extends an existing QSlim algorithm
- allows user-guidance to improve approximations
- little user effort, still efficient

Changes vertex tree structures

- can be used for further applications

Future Work

Better weights selection

- currently chosen by users
- automatic suggestion would be good

Perceptually based error metric

- low-level perception models

  (Reddy ‘97, *Luebke and Hallen ’01*)
- but also high-level : eg. actual feature vs noise
- machine learning techniques?

The End

Thanks!

Contact Information

Youngihn kho	*Michael Garland*
kho@uiuc.edu	*garland@uiuc.edu*